Journal of Artificial Intelligence Research 82 (2025) 1147-1173 Submitted 12/2024; published 02/2025

Enhanced Recommendation Systems with Retrieval-Augmented Large Language Model

Chuyuan Wei WEICHUYUAN@BUCEA.EDU.CN
College of Electrical and Information Engineering
Beijing University of Civil Engineering and Architecture
Beijing, China

Ke Duan 2108020022005@STU.BUCEA.EDU.CN
College of Mechanical-Electronic and Vehicle Engineering
Beijing University of Civil Engineering and Architecture
Beijing, China

Shengda Zhuo ZHUOSD96@GMAIL.COM
(Corresponding Author)
College of Cyber Security
Jinan University
Guangzhou, Guangdong, China

Hongchun Wang WANGHONGCHUN@BUCEA.EDU.CN
College of Urban Economics and Management
Beijing University of Civil Engineering and Architecture
Beijing, China

Shuqiang Huang HSQ@JNU.EDU.CN
(Corresponding Author)
College of Cyber Security
Jinan University
Guangzhou, Guangdong, China

Jie Liu LIUJXXXY@126.COM
North China University of Technology
Beijing, China

Abstract

Recommender systems have long struggled with challenges such as cold start and data sparsity, which can lead to poor recommendation performance. While previous approaches have attempted to address these issues by incorporating side information, they often introduce noise, lack flexibility for data expansion, and suffer from inconsistent data quality—factors that hinder accurate user preference inference and reduce recommendation performance. With the vast knowledge bases and advanced reasoning capabilities of large language models (LLMs), these models are particularly well-suited to supplement auxiliary information and capture implicit user intent. To address these challenges, we propose a novel framework, ER^{2}ALM, which leverages the capabilities of LLMs enhanced by Retrieval-Augmented Generation (RAG) to improve recommendation outcomes. Our framework specifically addresses the challenges by flexibly and accurately augmenting auxiliary information and capturing users' implicit preferences and interests. Additionally, to mitigate the risk of introducing noise, we incorporate a noise reduction strategy to ensure the reliability of the augmented information. Experimental validation on two real-world datasets demonstrates the efficacy of our approach, significantly enhancing both the ac-

curacy and robustness of recommendations compared to state-of-the-art methods. This demonstrates the potential of our framework as a new paradigm for preference mining in recommendation systems.

1. Introduction

With the rapid growth of the Internet, the issue of information overload (Zhuo et al., 2024b; Chen et al., 2022a; Tian et al., 2022; Wei et al., 2022; Zhou et al., 2023) has become increasingly urgent. Recommender systems can effectively mitigate this challenge by leveraging users' historical interactions to provide personalized and accurate recommendations from vast amounts of data. Understanding users' genuine interaction intentions (*e.g.,* preferences and interests) is crucial for recommendation accuracy, as users typically select items based on their individual needs and inclinations. These hidden user intentions, further mined from historical interaction data, can improve recommendation relevance.

User interaction histories serve as a record of past behaviors, inherently containing a wealth of user-specific information. The items within these histories may reflect products that users actively seek out due to interest or select purposefully to meet particular needs. This information, when cross-referenced with other relevant data, can provide a foundational basis for user analysis. For instance, in movie recommendation systems, users often provide ratings and reviews for the movies they watch. High ratings and positive reviews offer initial insights into implicit user preferences. Some studies have explored the use of side information as a means to extract sparse implicit feedback signals for enhancing user preference analysis. To address this, several approaches have integrated Graph Neural Networks (GNNs) into Collaborative Filtering (CF) frameworks (*e.g.*, NGCF (Wang et al., 2019), LightGCN (He et al., 2020)). Yet, these methods often encounter challenges stemming from insufficient supervisory signals. To mitigate this issue, recent studies (Ren et al., 2024b) have employed contrastive learning (*e.g.*, SGL (Wu et al., 2021), SimGCL (Yu et al., 2022)) techniques to enhance self-supervised signals. Real-world online platforms, such as Netflix and MovieLens, effectively leverage multi-modal content to enhance user experiences. In light of this, recent methodologies, as opposed to traditional collaborative filtering (CF) approaches (Le & Lauw, 2021), focus on integrating auxiliary side information to improve recommendation systems. For instance, MMGCN (Wei et al., 2019) and GRCN (Wei et al., 2020) incorporate item-side content into GNNs to uncover higher-order relationships that are informed by content. Similarly, LATTICE (Zhang et al., 2021) utilizes auxiliary content to perform data augmentation by establishing relationships between items. Recent innovations, including MMSSL (Wei et al., 2023) and MICRO (Zhang et al., 2022), tackle the challenge of data sparsity by implementing self-supervised tasks that maximize mutual information across various content-augmented views. When primary data is incomplete, the integration of auxiliary information may sometimes be unsuitable for inferring user characteristics. Although the primary intent behind introducing auxiliary data is to enrich product information, it often fails to enhance the system's ability to describe user-specific traits. Consequently, this approach may not only diminish the accuracy of user-item interaction analysis but also hinder the effective extraction of user preferences.

User interactions in recommendation systems are typically intention-driven, often reflecting the user's interests or preferences. Understanding such interests can be derived

from various forms of feedback, including interaction history, item attributes, and user reviews (Caro-Martínez et al., 2021). However, not all of this information contributes equally to improving recommendations: (1) Item attributes typically lack the depth necessary for accurately inferring user preferences; they provide general descriptions of items from different aspects but fail to capture the specific elements that appeal to individual users. (2) User reviews may obscure actual preferences, as incomplete or inaccurate feedback complicates the analysis. Thus, effectively extracting implicit user intentions from historical interaction sequences becomes critical for generating accurate personalized recommendations. Recent researches (He et al., 2024; Zhuo et al., 2024a; Chen et al., 2022b; Tanjim et al., 2020) has employed advanced deep learning methods to explore the underlying implicit intent behind user behavior. Recent efforts (Wu et al., 2023; Wei et al., 2024; Zheng et al., 2024) have attempted to leverage LLMs to extract user preferences more effectively. Supplementary auxiliary information can improve the effectiveness of user preference mining, however, an excessive increase in the amount of information may lead to diminished accuracy in capturing meaningful preferences.

Incorporating external knowledge to supplement the basic information about items and conducting analysis on this enhanced data is an effective approach for capturing implicit content. By providing additional attributes, external knowledge enriches the available information about the item, which in turn facilitates a more precise analysis of user intent. Fig. 1 demonstrates this limitation, where the minimal amount of core data on movies constrains the system's ability to derive meaningful insights. When supplemented with auxiliary information based on basic movie details, the system can identify shared features, such as the same language and director, across the three viewing histories. The expansion of auxiliary information significantly enhances the inference of user preferences and needs. Furthermore, relying solely on user reviews and ratings may fail to fully capture the complexity of user needs. Despite the advanced language understanding and reasoning capabilities of LLMs, accurately discerning genuine user needs from review data remains challenging, thereby making the effective utilization of such information significantly difficult. The integration of auxiliary information not only enhances the richness of the available data but also strengthens the foundation for analyzing user behavior and making informed recommendations. However, the effectiveness of the recommendation process is contingent on the quality of this expanded information. A significant challenge remains in mitigating the noise introduced by low-quality data during the expansion, leaving the effective utilization of such data an open question.

To address the aforementioned challenges, we propose an Enhanced Recommendation Systems with Retrieval-Augmented Large Language Model (ER^{2}ALM) approach tailored for recommendation. We address this issue by leveraging the user's reliable interaction history with items, using highly-rated items as key indicators of user preferences. By focusing on content that users favor, we mitigate the noise introduced by less-preferred items. It is crucial to recognize that user reviews often reflect emotional responses rather than genuine needs, leading to potential misinterpretations; hence, reviews are excluded from further analysis. By extending the attributes of each item within the user's interaction history, we focus on uncovering patterns or hidden connections in their selection behavior. Using LLMs, we extract deeper insights into the user's preferences by identifying subtler associations. During the item auxiliary information expansion process, we integrate RAG to

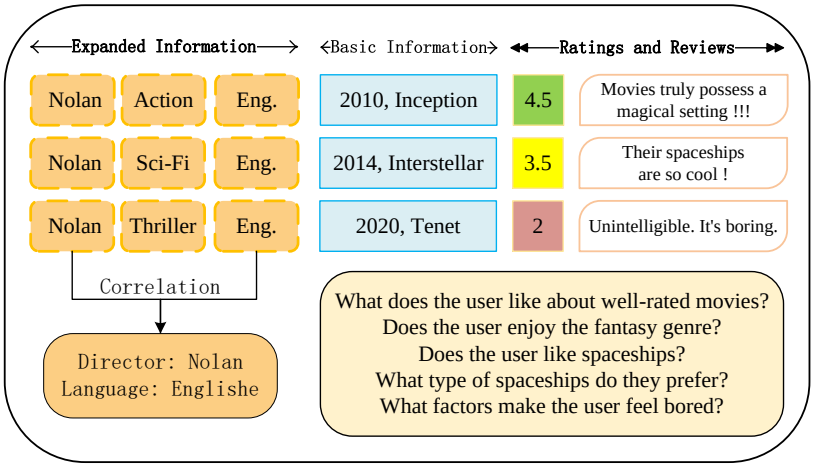

Figure 1: Viewing history can uncover potential connections through extended information, whereas relying solely on user ratings and reviews makes it challenging to gather valuable feedback and limits the understanding of users' preferences. Unable to answer the questions in the bellow box accurately makes it difficult to trust the review ratings.

ensure that the additional data is both relevant and accurate. This method, in combination with LLMs, strengthens the basis for user profile analysis and recommendation generation. Given the possibility of errors or instability in LLM-generated content, we employ a denoising mechanism. This mechanism leverages graph-based correlations to evaluate the reliability of generated content, retaining only the most trustworthy embeddings for training while filtering out noisy data.

In summary, our **contributions** can be outlined as follows:

- ER^{2}ALM integrates RAG and LLMs to effectively augment item auxiliary information, resulting in more accurate user modeling.

- ER^{2}ALM generates personalized user profiles from scratch using LLMs, providing enhanced personalization and boosting model performance.

- Extensive evaluations on real-world datasets demonstrate that ER^{2}ALM outperforms state-of-the-art baseline methods, establishing its superiority.

The remainder of this paper is organized as follows: Section 2 reviews related work, while Section 3 details the proposed methodology. In Section 4, we present extensive experiments and analyze the corresponding results. Finally, Section 5 concludes the paper and discusses potential future work.

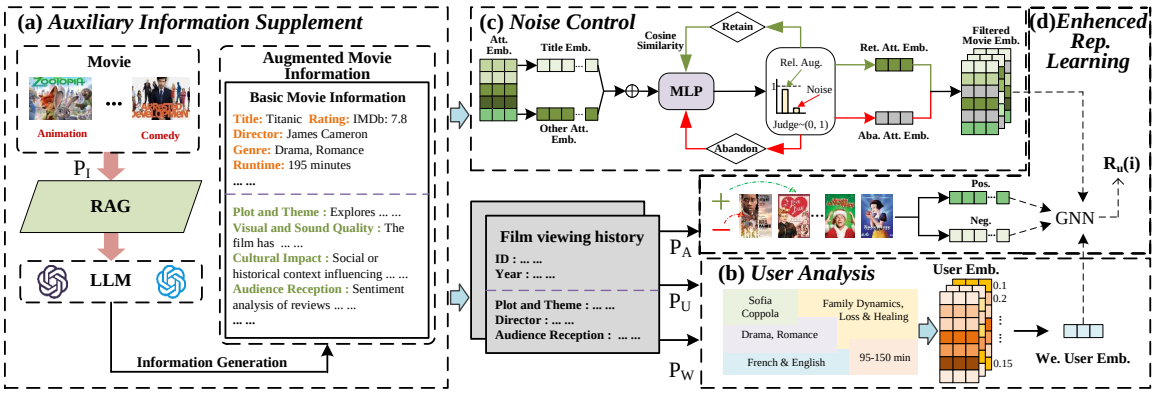

Figure 2: The overall framework of the proposed ER^{2}ALM, which consists of four modules: (a) Auxiliary Information Supplement, (b) User Analysis, (c) Noise Control, (d) Enhanced Representation Learning. RAG represents Retrieval-Augmented Generation and LLM represents Large Language models. (Transf., Att., Emb., Ret., Aba., Rep., We., are the short names for Transforms, Attribute, Embedding, Retain, Abandon, Representation, Weights.)

2. Related Work

2.1 Large Language Models for Recommendation

LLMs are the scaled-up derivatives of traditional pretrained language models, in terms of both model sizes and data volumes. Several studies have incorporated LLMs as a component in recommendation algorithms. For instance, in UniSRec (Hou et al., 2022), a fixed BERT model is used to learn item representations, which are then enhanced via a lightweight MoE-based network to facilitate cross-domain sequential recommendations. Building on UniSRec, VQ-Rec (Hou et al., 2023) introduces vector quantization techniques to better align the text embeddings generated by LLMs with the recommendation space. Uni-CTR (Fu et al., 2023) leverages hierarchical semantic representations in shared LLMs to capture commonalities across different domains, thereby improving multi-domain recommendation performance. Zhiyuli et al. propose a method where LLMs predict user ratings in a text-based manner. Chat-REC (Gao et al., 2023) employs ChatGPT to bridge the gap between conversational interfaces and traditional recommender systems by modifying the returned item candidates before presenting them to users. LLaRA (Liao et al., 2024) adopts a novel hybrid approach to represent items in the LLM input prompt, combining ID-based item embeddings from traditional tokenizers with textual item features. SINGLE (Liu et al., 2024b) leverages LLMs to capture the user's constant preferences from historical interactions, while employing contrastive learning techniques to capture instantaneous preferences. CLLM4Rec (Zhu et al., 2024) utilizes a mutual regularization strategy through soft+hard prompting during pretraining, effectively capturing both collaborative and content-based information of users and items via language modeling. In comparison, our model leverages the generative and robust text comprehension capabilities of the LLMs for inference, while mitigating the issue of excessive dimensionality that typically arises when LLMs are used

purely as encoder. By avoiding the direct use of LLMs in the recommendation decision-making process, our approach mitigates the discrepancies between the recommendation task and the characteristics of LLMs.

2.2 Retrieval-Augmented Generation for Recommendation

RAG is a technique that enhances text generation quality by retrieving precise and comprehensive information from external databases, which is then applied to downstream tasks. Recently, retrieval-augmented large language models (RA-LLMs) have demonstrated considerable promise in delivering personalized and contextually relevant recommendations by effectively integrating retrieval and generation processes (Di Palma, 2023; Wu et al., 2024). For example, Di Palma (Di Palma, 2023) presents a straightforward retrieval-augmented recommendation model that utilizes knowledge from movie and book datasets to enhance recommendation quality. Additionally, Lu et al. (Lu et al., 2021) improve the retrieval process by incorporating user reviews, thereby enriching item information within recommender systems. CoRAL (Wu et al., 2024) leverages reinforcement learning to extract collaborative information from datasets, aligning it with semantic information for more precise recommendations. RaSeRec (Zhao et al., 2025) utilizes memory retrieval to extract collaborative memories for the target user, effectively adapting to dynamic preference shifts through a real-time updating memory bank. From the perspective of understanding and uncovering users' latent preferences, we leverage the advanced text generation capabilities of LLMs to retrieve more complex and comprehensive information from specialized databases, thereby enriching the model with additional, multidimensional contextual information.

2.3 Data Augmentation for Large Language Models

LLMs, with their open-world knowledge, can be viewed as flexible knowledge repositories, providing auxiliary functions for user preference modeling and item content understanding. For example, KAR (Xi et al., 2024) leverages LLMs to generate user-side preference knowledge and item-side factual knowledge, which serve as additional features for downstream recommendation models. SAGCN (Liu et al., 2024a) introduces a chain-based prompting method to uncover semantically-aware interactions, offering clearer insights into user behaviors. CUP (Torbati et al., 2023) employs ChatGPT to analyze user review texts and summarize each user's interests using a few concise keywords. Additionally, LLaMA-E (Shi et al., 2024) and EcomGPT (Li et al., 2024) enhance various downstream generation tasks in e-commerce scenarios using large language models, such as product classification and intent inference. Other studies further enrich training data through LLMs from different perspectives, including attribute generation (Brinkmann et al., 2023; Li et al., 2023; Yin et al., 2023) and user interest modeling (Christakopoulou et al., 2023; Doddapaneni et al., 2024; Lyu et al., 2024; Ren et al., 2024a). Building upon the aforementioned methods, we have leveraged the flexibility and convenience of LLMs to expand the available information. Additionally, by incorporating RAG, we ensure the retrieval of reliable and relevant information, thereby enhancing the reliability of the generated auxiliary data.

Table 1: Hard prompt structure and content description

Prompt Section	Purpose
Task Description	Provides a concise explanation of the task, clearly defining the objectives and expected outcomes.
Task Instructions	Outlines the specific input requirements and describes the approach or steps necessary to complete the task.
Specific Detail Requirements	Specifies the structure of the output and highlights key content areas that must be addressed in the response.
Output Format Specifications	Defines both the input and output data fields, clarifying their content, significance, and intended usage.
Emphasis on Details	Draws attention to key considerations and instructions to prevent errors or misinterpretation.

3. Proposed Approach

This section provides a formal definition of the recommendation problem and introduces the proposed ER^{2}ALM methodology, which is made up of five core modules: (1) *Auxiliary Information Supplement*: Integrates RAG and LLMs to retrieve relevant auxiliary data, thereby enhancing recommendation accuracy. (2) *User Analysis*: Employ LLMs to analyze user behavior and construct personalized profiles. (3) *Embedding Generation*: Using the RoBERTa model to transform text into high-quality embeddings. (4) *Noise Control*: Mitigates noise to preserve the quality and reliability of embeddings. (5) *Enhanced Representation Learning*: Processes the refined data to facilitate effective model training. By incorporating user preferences alongside detailed historical data, our model significantly improves its capacity to capture the underlying recommendation logic. Fig. 2 provides an overview of the complete framework.

Problem Definition. Let $U = \{u_1, u_2, \ldots, u_{|U|}\}$ represent the user set, and $I = \{i_1, i_2, \ldots, i_{|I|}\}$ represent the item set, where $|U|$ and $|I|$ denote the number of users and items, respectively. Specifically, let $\boldsymbol{i}_n = \{\text{basic}_{i,n} \mid \boldsymbol{i}_n \in I, n = 1, 2, \ldots, |I|\}$ represent the basic attributes of each item. Moreover, let $\boldsymbol{i}_n^A = \{(\text{basic}_{i,n}, \text{auxil.}_{i,n}) \mid \boldsymbol{i}_n^A \in I^A, n = 1, 2, \ldots, |I|\}$ denote the auxiliary information expansion for each item, where I^A represents the set of items enriched with auxiliary information. For a given user $\boldsymbol{u}_m \in U$, let $\boldsymbol{h}_m = \{i_{m,1}^A, i_{m,2}^A, \ldots, i_{m,|h_m|}^A\}$ represent the historical sequence of the user, where $|h_m|$ denotes the length of the sequence and $h_m \in H_U$ represents the set of user interaction histories. The user profile $\mathbf{q}_m = \{\text{prefe. analy.}_{m,k} \mid \mathbf{u}_m \in U, m = 1, 2, \ldots, |U|\}$, with $q_m \in Q_U$ denoting the set of user profiles, and the corresponding preference weights $\boldsymbol{w}_{m,k} \in W_U$ representing the set of profile weights, are derived by analyzing the interaction history sequence h_m, where k represents the number of categories in the analysis process. User profile embeddings and item embeddings are denoted as $\mathbf{e}_m^q$ and $\mathbf{e}_n^i$, respectively. Positive samples (E_i^+) represent items that are likely to engage users, while negative samples (E_i^-) denote those that are less likely. The model predicts interaction scores as $\hat{r}(u, i)$ after training.

3.1 Auxiliary Information Supplement

To infer user preferences based on the aggregation and analysis of item attributes, it is first necessary to augment incomplete attribute information. To obtain accurate, reliable, and content-rich auxiliary information, we propose a strategy that integrates RAG with LLMs to supplement missing data.

By leveraging RAG techniques, more detailed information about an item can be retrieved based on its name or category. Let $\boldsymbol{i}_n = \{\text{basic}_{i,n} \mid \boldsymbol{i}_n \in I, n = 1, 2, \ldots, |I|\}$ represent the basic attributes of each item. After completing the acquisition of external item information, we employ the LLMs to perform the generative process:

$$\boldsymbol{i}_n^A = L(P_I(\boldsymbol{i}_n, A[R(\boldsymbol{i}_n)])), n = 1, 2, ..., |I|, \tag{1}$$

where $L(\cdot)$ denote the LLMs, $R(\cdot)$ and $A(\cdot)$ represent the search and augmentation components of the RAG framework, respectively, and P_I stands for the prompts used to generate the outputs, with its structure detailed in Table 1.

The retrieved information can be categorized into three types: essential and directly usable information, redundant or irrelevant information, and complex, fragmented information requiring refinement and summarization. Leveraging the advanced language understanding and text generation capabilities of LLMs, we further extract valuable insights from the retrieved data, enabling targeted augmentation of auxiliary information.

Let $\boldsymbol{i}_n^A = \{(\text{basic}_{i,n}, \text{auxil.}_{i,n}) \mid \boldsymbol{i}_n^A \in I^A, n = 1, 2, \ldots, |I|\}$ denote the auxiliary information expansion for each item, where I^A represents the set of items enriched with auxiliary information. The values $\text{basic}_{i,n}$ remain the same as in the base item information.

This method mitigates the limitations of LLMs, particularly in generating incomplete or erroneous content, and minimizes the risk of producing unreliable 'phantom' responses. The application of LLMs for content generation is not only straightforward but also highly adaptable, enabling the rapid creation of supplementary information.

3.2 User Analysis

Subsequently, we strategically expanded the auxiliary information associated with items to foster a deeper understanding of the items and establish a robust foundation for utilizing item-related data to infer user preferences. A user's interaction history is assumed to reflect their preferences, as it consists of items they found particularly satisfying. By analyzing this enriched interaction history, the underlying intentions and preferences guiding the user's item selection can be inferred. Integrating the comprehensive purchase history with predesigned prompt templates enables the model to perform precise reasoning. This process identifies key decision points and uncovers logical relationships, facilitating the extraction of implicit user intentions and interests to construct a detailed and accurate user profile.

We use a fixed template of prompts delivered to the LLMs in the form of hard prompts. Inspired by the ideas presented in (Fatemi et al., 2024), we detail the data content and interpretive approach for user interaction records and relevant product information, further enhancing the generative capabilities of LLMs,

$$\boldsymbol{q}_m = L\{P_U[\boldsymbol{h}_m, \underbrace{S(\mathbf{i}_n^A)}_{n \in h_{m,1}, h_{m,2}, \ldots, h_{m,|h_m|}}, Temp]\}, \tag{2}$$

$$Temp = \{(T_{data}), Des, Ins, Det, Out, Emp\}, \tag{3}$$

where $q_m \in Q_U$ denoting the set of user profiles, $\boldsymbol{h}_m$ represents the user's interaction history, $S(\cdot)$ denotes the collection of interaction histories enriched with relevant product information, and *Temp* refers to the hard prompt template. P_U represents the hard prompt used to guide the analysis and generate the user profile, while $L(\cdot)$ denotes the LLMs. Together, these components form the cue information for analyzing user preferences, which is processed by the LLMs to infer the user's interests. The hard prompt template *Temp* consists of key components, detailed in Table 1. This structure helps mitigate common errors in LLMs processing, and the prompt structure in the method follows this format.

Next, user profiles are derived by analyzing users from multiple perspectives, with each perspective closely aligned to the classifications used in the auxiliary information generation process. These perspectives highlight the key factors users prioritize when selecting items. Additionally, the degree of emphasis on each perspective varies across users. To capture this variability, distinct weights are assigned to represent the influence of each perspective, further optimizing the personalized representation of users. Building upon the analysis of user interaction history and preferences, we further employ LLMs to analyze and derive personalized preference weights for each user. This detailed analysis facilitates a deeper understanding of user behavior, enabling the extraction of implicit intentions and interests to support personalized user modeling. Fig. 3 illustrates the prompts used for generating user profiles, along with their corresponding outputs.

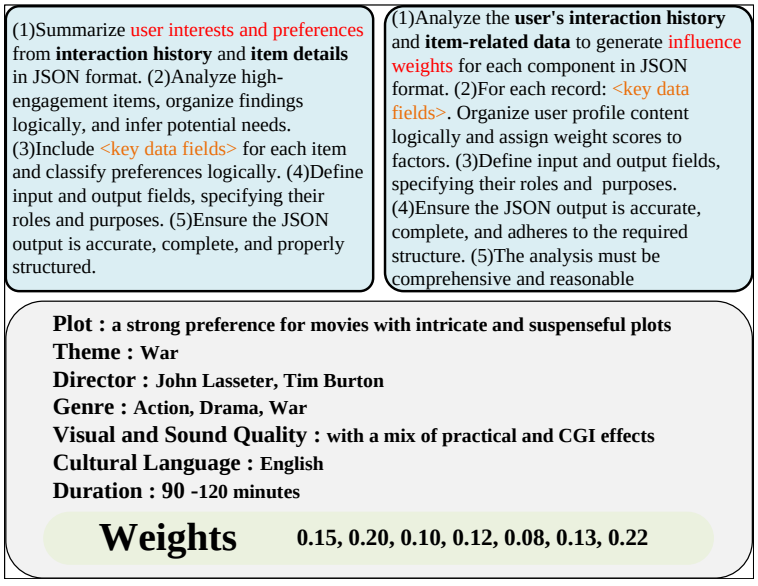

Figure 3: The top part shows the inputs for prompts $\mathbf{Prompt_U}$ and $\mathbf{Prompt_W}$, while the bottom part corresponds to the output.The $< \cdot >$ indicates where specific data entries have been omitted.

3.3 Embedding Generation

In continuation, we utilize the RoBERTa model to convert textual information (*e.g.* generated text and user profiles), into embeddings. For item-related data, each attribute is individually embedded. A user profile is represented as a composite embedding, which is formed by combining various aspects according to their importance weights, thereby reflecting the user's prioritization of different content elements. Specifically, during the embedding process, these weights are applied to merge multiple pieces of information into a unified embedding, with each piece contributing based on its relative importance. This approach ensures that more important data have a proportionally greater influence on the overall representation, thereby preserving relevant information while highlighting its varying degrees of significance. By capturing both the breadth and depth of user preferences, this method enhances the model's capacity to reflect the subtleties of user behavior.

The user profile vector $\mathbf{e}_m^q$ is a weighted sum of siveral aspects:

$$\boldsymbol{e}_n^i = \{e_1^i, e_2^i, \ldots, e_{k'}^i\}, \quad n \in [1, |I|], \tag{4}$$

$$\boldsymbol{e}_m^q = \sum_{i=1}^{k} q_{m,k} w_{m,k}, \quad w_{m,k} \in (0,1), \quad m \in [1, |M|], \tag{5}$$

where $e_{k'}^i$ represents the k'-th aspect of item e_n^i, $q_{m,k}$ represents the k-th aspect of user profile q_m, and $w_{m,k}$ are the corresponding weights. Specifically, we denote the embeddings of the user profiles and the items as e_m^q and e_n^i, respectively, where $\boldsymbol{e}_m^q, \boldsymbol{e}_n^i \in \mathbb{R}^{1\times d}$, d is the dimension of the user profile embeddings and the item embeddings, where $|M|$ and $|I|$ denote the number of users and items, respectively.

If embeddings are generated using LLMs, several challenges may arise, including excessive embedding length and inconsistencies in dimensionality across different text embeddings. By employing a traditional text conversion approach, we can effectively manage the embedding length while maintaining the differentiation of content elements, thereby ensuring that no critical information is lost during the dimensionality reduction process.

3.4 Noise Control

When enriching auxiliary information through LLMs, some incorrect or irrelevant text may inevitably be generated, introducing noise into subsequent training. To mitigate this, we adopt an efficient method for validating auxiliary information after the embedding process. First, we select one or several basic attributes as trustworthy markers, which serve as reliable references for evaluation. We then compute the correlation between their embeddings and other embeddings using a similarity metric.

To achieve this, we utilize the graph destruction method (Wang et al., 2023) to calculate the correlation between the embeddings of the selected markers and each piece of generated information. The resulting correlation values range from 0 to 1, where higher values indicate greater reliability. Based on these values, we decide whether to retain or discard the auxiliary information, ensuring that only reliable data are preserved for subsequent analysis.

$$w_{n,k'}^{i^A} = \mathrm{MLP}\left[W_r\left(\mathbf{basic}_i \mid i_{n,k'}^A\right)\right], n \in (1, 2, ..., |I|), \tag{6}$$

where $w_{n,k'}^{i^A}$ represents the edge weight of the two attributes, $i \in I$, $i^A \in I^A$, MLP$(\cdot)$ is short for multi-layer perception, and $W_r(\cdot)$ is the transformation matrix of relation r. Additionally, basic$_i$ and $i_{n,k'}^A$ denote the embeddings of the basic attributes and other generated attribute for each item, respectively. k' represents the k'-th aspect of item i_n^A. I denotes the item set, and I^A denotes the augmented item set. $|I|$ denotes the number of items,

$$\epsilon = (2 \times bias - 1) \times \text{rand(d)} + (1 - bias), \tag{7}$$

where ϵ represents a randomly generated offset, while the *bias* is a small value introduced to prevent issues of complete rounding or full retention, ensuring numerical stability in the computations. And d is the dimension of the item embeddings.

$$J_{n,k'}^{i^A} = \sigma\left(\left(\log(\epsilon) - \log(1-\epsilon) + w_{n,k'}^{i^A}\right)/\tau_{m^A}\right), i^A \in I^A, n \in (1,2,...,|I^A|), \epsilon \in (0,1), \tag{8}$$

where the random variable ϵ is drawn from the range (0, 1), $\sigma(\cdot)$ represents the Sigmoid function, and the temperature hyperparameter τ_{m^A} controls the approximation. τ_{m^A} approaches 0, $J_{n,k'}^{i^A}$ will tend toward binary values. θ denotes the threshold for determining accuracy or inaccuracy. When $J_{n,k'}^{i^A} > \theta$, embedding will be retained as reliably generated, otherwise embedding will be discarded as low-quality information.

Based on the aforementioned classification method, we can collect both correct and incorrect classification results. These classified outcomes are stored for subsequent evaluation. Once a sufficient number of reliable and unreliable results have been categorized, new content is compared against these stored embeddings. By calculating cosine similarity, we can assess whether the new content is more closely aligned with reliable or unreliable results, thereby guiding the decision to retain or discard it.

We maintain two sets of embeddings: one for reliable data and one for unreliable data. When an embedding's confidence score lies between the unreliable and reliable thresholds, cosine similarity is used to compare it against the embeddings in both sets. The category with the highest similarity score is selected as the final classification outcome. Once the number of embeddings in the two categories reaches a certain threshold, the to-be-judged augmented attribute embedding $\mathbf{E}_A^J$ can be expressed as follows:

$$\mathbf{E}_A^J = \begin{cases} \mathbf{Emb}_{n,k'}^{i^A}, & J_{n,k'}^{i^A} >> \theta_{\text{reliable}}, \\ \mathbf{Emb}_{n,k'}^{i^A}, & J_{n,k'}^{i^A} >= \theta_{\text{reliable}} \text{ and } \Delta > 0, \\ \emptyset, & J_{i,k'}^{i^A} < \theta_{\text{reliable}} \text{ or } \Delta < 0, \end{cases} \tag{9}$$
$$i^A \in I^A,\ n \in (1,2,...,|I^A|),\ \epsilon \in (0,1),$$

for high values of $J_{n,k'}^{i^A}$ we determine that the information quality is reliable. When the score is low, cosine similarity is used to assess reliability. Let θ_{reliable} represent the threshold for reliability. k' represents the k'-th aspect of item i_n^A,

$$\Delta = \mathbf{sim}(\mathbf{E}_{retain}, \mathbf{Emb}_A) - \mathbf{sim}(\mathbf{E}_{abandon}, \mathbf{Emb}_A), \tag{10}$$

where the variable Δ represents the difference between the similarity to the retain and abandon embeddings, $\mathbf{E}_{retain}$ and $\mathbf{E}_{abandon}$. The variable $\mathbf{Emb}_A$ indicates the embedding

Algorithm 1 Training Procedure of ER^{2}ALM

Input: Item set I, User set U, Historical interaction set H_U
Output: Top-k Recommendations

for each item $i \in I$ **do**
  $i^A \leftarrow \text{RAG}(i) + \text{LLM}(i)$ via Eq.1
end for
for each user $u \in U$ **do**
  $h_u \leftarrow H_U$
  $Profile_u \leftarrow \text{Analyze}(h_u, i^A)$ via Eq.2
  ${u_i}^+ \leftarrow \text{LLM}(u[\text{pos}])$
  ${u_i}^- \leftarrow \text{LLM}(u[\text{neg}])$
  $e^u \leftarrow \text{RoBERTa}(Profile_u)$ via Eq.5
  $E_i^+, E_i^- \leftarrow \text{RoBERTa}({u_i}^+, {u_i}^-)$
end for
for each augmented item $i^A \in I^A$ **do**
  $e^i_{k'} \leftarrow \text{RoBERTa}(i^A)$
  for each attribute embedding $e^i_{k'}$ of i^A **do**
    $J_A \leftarrow \text{EvaluateReliability}(e_{basic}, e_{k'})$ via Eq.8
    if $|\mathbf{E}_{retain}| < N$ or $|\mathbf{E}_{abandon}| < N$ **then**
      if $J_A \geq \theta_{\text{reliable}}$ **then**
        Add $e^i_{k'}$ to the reliable set $\mathbf{E}_{retain}$
      else
        Add $e^i_{k'}$ to the unreliable set $\mathbf{E}_{abandon}$
      end if
    else
      Compute similarity difference Δ via Eq.10
      if $J_A \geq \theta_{\text{reliable}}$ and $\Delta > 0$ **then**
        Confirm $e^i_{k'}$ as reliable and retain it
      else
        Discard $e^i_{k'}$ as unreliable
      end if
    end if
  end for
end for
Compute interaction scores for all user-item pairs via Eq.13
Select top-k items with highest $\hat{r}(u, i)$ for each user
return $R_u(i), \quad \forall u \in U, \quad \forall i \in I$

to be compared against both reliable and unreliable categories. If $\Delta > 0$, the information is considered reliable; otherwise, it is deemed unreliable.

The cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$ is computed as follows:

$$\mathbf{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}, \tag{11}$$

where $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ are the norms of the vectors $\mathbf{u}$ and $\mathbf{v}$, respectively. If the new content is determined to be reliable, its embedding vector is appended to the existing embedding matrix.

The loss function *Loss* for distinguishing between accurate and inaccurate product information is computed as:

$$Loss = -\sum_{i \in |E|} \log \sigma(\mathbf{Emb}_A \cdot \mathbf{E}), \tag{12}$$

where $\sigma(\cdot)$ is the sigmoid function, $\mathbf{Emb}_A$ denotes the embedding vector of the auxiliary product information being evaluated, while $\mathbf{E}$ refers to the embedding vector of the selected reliable attributes, which serve as the reference for comparison.

3.5 Enhanced Representation Learning

Traditional recommendation models generate recommendation lists based on users' historical browsing records and corresponding training results (Caro-Martínez et al., 2021; Burashnikova et al., 2021; Koto et al., 2022). By leveraging the decision-making and comprehension capabilities of LLMs, we can further enhance this process. From the generated limited recommendation list, LLMs selects the items that users are most likely to be interested in and relatively uninterested in as positive and negative samples (E_i^+, E_i^-). Using a small number of generated positive and negative samples in the training process can increase the interactive information and form a certain degree of comparison.

Finally, the generated user profile, along with the embeddings of the filtered auxiliary information and a subset of the identified positive and negative samples, are fed into the GNNs model for training. This process leads to the final predictive model, which is optimized to provide more accurate and personalized recommendations. Additionally, we calculate the probability that user u will interact item i using the inner product as follows:

$$\hat{r}(u, i) = \mathbf{e}_u^T \mathbf{e}_i, \tag{13}$$

where $\mathbf{e}_u$ and $\mathbf{e}_i$ represent the embeddings of the user and item, respectively. Additionally, we adopt the BPR loss function to calculate the loss for our task, whose formulation can be defined as:

$$Loss = \sum_{(i,i') \in S} -\log \sigma(\hat{r}(u, i) - \hat{r}(u, i')), \tag{14}$$

where $\sigma(\cdot)$ is the igmoid function, S denotes the test set, i denotes the item in the interaction history, and i' denotes the item not in the user history.

4. Experiment

In this section, we provide empirical evidence demonstrating the feasibility and effectiveness of the proposed ER^{2}ALM algorithm in personalized recommendation systems. The algorithm effectively leverages auxiliary data to mine user preferences and seamlessly integrates this information into the recommendation process. To validate its performance, we conducted experiments in the movie recommendation domain, utilizing two widely adopted

Table 2: Statistics of the Original and Augmented Datasets

Dataset		**Netflix**			**MovieLens**		
Graph	Ori.	**# U**	**# I**	**# E**	**# U**	**# I**	**# E**
		13,187	17,366	68,933	12,237	10,681	78,880
	Aug.	# E: $\lceil 26,374 * \eta \rceil$			# E: $\lceil 24,474 * \eta \rceil$		
Ori. Sparsity		99.970%			99.930%		
Att.	Ori.	U: *None*	I: year, title		U: *None*	I: title, year, genre	
	Aug.	**I[1024]: plot and theme, genre, visual sound characteristics, culture and language, movie duration, director.**					
		U[1024]: plot, theme, genre, visual sound characteristics, culture and language, movie duration, director. Weights.					
Modality		**Textual** [768]**, Visual** [512]			**Textual** [768]**, Visual** [512]		

* *Att.* represents attribute, *Ori.* represents original, and *Aug.* represents augmentation. The number in [X] represents the feature dimensionality. $\lceil x \rceil$ is the ceiling function and η is a coefficient.

movie datasets for evaluation. Section 4.1 outlines the general experimental settings, while Sections 4.2 to 4.4 provide results and findings, and Section 4.5 presents a comprehensive analysis of the case study. This section's experiments aim to answer the following research questions (**RQs**):

- **RQ1:** How does our LLM-enhanced recommender compare against the leading baselines in the field?
- **RQ2:** How do key components impact the overall performance of the model?
- **RQ3:** How sensitive is the model to variations in key parameters?
- **RQ4:** Can our model provide intuitive explanations for the prediction results?

4.1 Experimental Settings

In this part, we provide a concise overview of the general experimental setup, including details on the datasets, evaluation protocols, comparative baselines, and implementation specifics.

Datasets. We conduct experiments using publicly available datasets, Netflix and MovieLens-10M (ML-10M), both of which contain basic information about the movies. **Netflix**[1] released by Netflix, contains over 100 million anonymous movie ratings collected from users

1. https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data

Table 3: Performance comparison on different datasets in terms of Recall@10/20/50, NDCG@10/20/50, and Precision@20

Baseline		Netflix							MovieLens						
		R@10	**N@10**	**R@20**	**N@20**	**R@50**	**N@50**	**P@20**	**R@10**	**N@10**	**R@20**	**N@20**	**R@50**	**N@50**	**P@20**
BPR-MF	**CF**	0.0282	0.0140	0.0542	0.0205	0.0932	0.0281	0.0027	0.1890	0.0815	0.2564	0.0985	0.3442	0.1161	0.0128
NGCF		0.0347	0.0161	0.0699	0.0235	0.1092	0.0336	0.0032	0.2084	0.0886	0.2926	0.1100	0.4262	0.1362	0.0146
LightGCN		0.0352	0.0160	0.0701	0.0238	0.1125	0.0339	0.0032	0.1994	0.0837	0.2660	0.1005	0.3692	0.1209	0.0133
VBPR	**SI**	0.0325	0.0142	0.0553	0.0199	0.1024	0.0291	0.0028	0.2144	0.0929	0.2980	0.1142	0.4076	0.1361	0.0149
MMGCN		0.0363	0.0174	0.0699	0.0249	0.1164	0.0342	0.0033	0.2314	0.1097	0.2856	0.1233	0.4282	0.1514	0.0147
GRCN		0.0379	0.0192	0.0706	0.0257	0.1148	0.0358	0.0035	0.2384	0.1040	0.3130	0.1236	0.4532	0.1516	0.0150
LATTICE	**DA**	0.0433	0.0181	0.0737	0.0259	0.1301	0.0370	0.0036	0.2116	0.0955	0.3454	0.1268	0.4667	0.1479	0.0167
LLMRec		0.0531	0.0272	0.0829	0.0347	0.1382	0.0456	0.0041	0.2603	0.1250	0.3643	0.1628	0.5281	0.1901	0.0186
MICRO	**SSL**	0.0466	0.0196	0.0764	0.0271	0.1306	0.0378	0.0038	0.2150	0.1131	0.3461	0.1468	0.4898	0.1743	0.0175
CLCRec		0.0428	0.0217	0.0607	0.0262	0.0981	0.0335	0.0030	0.2266	0.0971	0.3164	0.1198	0.4488	0.1459	0.0158
MMSSL		0.0455	0.0224	0.0743	0.0287	0.1257	0.0383	0.0037	0.2482	0.1113	0.3354	0.1310	0.4814	0.1616	0.0170
ER^{2}ALM		**0.0566**	**0.0283**	**0.0840**	**0.0367**	**0.1469**	**0.0482**	**0.0042**	**0.2645**	**0.1376**	**0.3891**	**0.1767**	**0.6036**	**0.2165**	**0.0210**
Improve		**6.60%**	**4.04%**	**1.31%**	**5.76%**	**6.29%**	**5.70%**	**2.38%**	**1.61%**	**10.08%**	**6.81%**	**8.54%**	**14.30%**	**16.01%**	**12.09%**

between 1999 and 2005. **MovieLens** is a widely used series of benchmark datasets in recommendation system tasks. ML-10M[2] contains 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. Both datasets use only metadata, such as the title, genre, and year of the film, to simulate the absence of attribute information. Table 2 illustrates the statistical details of the original and augmented datasets for the user and project domains.

Evaluation Protocols. To evaluate the performance of our approach, we employ three standard metrics:

- **Recall@N**, which measures the proportion of relevant items retrieved among the top K recommendations.
- **(NDCG@N)**, which accounts for the ranking position of relevant items in the top K.
- **Precision@N**, which indicates the proportion of relevant items within the top K recommendations.

To mitigate potential biases in the test sampling process, we adopt the all-ranking evaluation strategy (Wei et al., 2021a, 2020). The results are averaged over five independent runs, with K set to 10, 20, and 50. Higher Recall@K values indicate better retrieval effectiveness, NDCG@K captures the ranking quality by weighting relevant items more heavily at higher ranks, and Precision@K reflects the concentration of relevant items within the top K recommendations.

Baselines. To comprehensively assess the performance of our approach, we compare it with four distinct categories of baseline methods, each addressing unique challenges and offering varied perspectives on recommendation tasks.

2. https://grouplens.org/datasets/movielens/10m/

- **General Collaborative Filtering (CF)**: This category includes traditional CF-based methods that rely solely on user-item interaction data for generating recommendations. These methods focus on learning user and item embeddings from interaction history without the incorporation of additional side information. Representative methods in this category include: **BPR-MF** (Rendle et al., 2012): A matrix factorization model optimized for ranking-based recommendations using Bayesian Personalized Ranking, **NGCF** (Wang et al., 2019): A Neural Graph Collaborative Filtering model that integrates graph neural networks to capture complex relationships in user-item interactions, and **LightGCN** (He et al., 2020): A simplified and efficient graph convolutional model that enhances recommendation accuracy by reducing unnecessary layers and transformations;

- **Side Information (SI)**: These methods incorporate auxiliary information, such as item attributes or multimodal data, to enhance the modeling of user preferences and item characteristics. By leveraging side information, these approaches address data sparsity and provide enriched user-item interaction insights. Key methods in this category include: **VBPR** (He & McAuley, 2016): A model that combines visual features with CF, allowing for recommendations informed by the visual appeal of items, **MMGCN** (Wei et al., 2019): This method integrates multimodal information, such as images and text, into graph convolutional networks for a more comprehensive user representation, and **GRCN** (Wei et al., 2020): This model incorporates content-based features, such as tags or genres, to capture semantic relationships between items in a GCN framework.

- **Data Augmentation (DA)**: These approaches generate additional training data or enhance existing data to improve model robustness. These methods address data sparsity and create more diverse user and item representations. Notable examples include: **LATTICE** (Zhang et al., 2021), which enriches user-item data by uncovering latent item relationships, improving the model's ability to make recommendations in sparse datasets, and **LLMRec** (Wei et al., 2024): This approach applies large language models to enrich user-item data, providing context and additional descriptive information, especially useful in text-heavy domains.

- **Self-supervised Learning (SSL)**: These methods enhance representation quality by introducing auxiliary self-supervised tasks that generate pseudo-labels from interaction data, eliminating the need for manual labeling. These tasks help models learn more expressive representations through contrastive learning objectives. The main approaches in this category include: **CLCRec** (Wei et al., 2021b): A model that applies contrastive learning to CF by generating augmented samples, enhancing the robustness of user and item embeddings, **MMSSL** (Wei et al., 2023): This method maximizes mutual information between multimodal data views, enhancing user representations by leveraging diverse content types, and **MICRO** (Zhang et al., 2022): By aligning multiple augmented views of user-item interactions, MICRO improves recommendation quality, especially in sparse environments.

Implementation Details. This study employs the locally deployed ChatGLM3-6B[3] model to enhance data through LLM-generated dialogs. The AdamW optimizer (Paszke et al.,

3. https://huggingface.co/THUDM/chatglm3-6b

2019) was employed for training, with learning rates ranging from $[5 \times 10^{-5}, 1 \times 10^{-3}]$ for the Netflix dataset and $[2.5 \times 10^{-4}, 9.5 \times 10^{-4}]$ for the MovieLens dataset. For the LLMs parameters, the temperature was selected from $\{0.4, 0.8, 1\}$, aiming to balance the accuracy and richness of the generated content. The top-p value, used to control generation precision, was chosen from $\{0.6, 0.8, 1\}$. To maintain response integrity, data flow was disabled. For embedding generation, we utilized a 1024-dimensional RoBERTa model to capture more detailed content. For noise reduction, the threshold was set to 0.4, with similarity judgments for distress added once the number of trusted embeddings reached 500.

4.2 Performance Comparison (RQ1)

Table 3 presents a comparison of our proposed ER^{2}ALM method against various baseline models. On the Netflix dataset, our model achieves a Precision@20 of 0.0042, reflecting a 2.38% improvement over the best-performing baseline. For NDCG@10, 20, and 50, our model achieves values of 0.0283, 0.0367, and 0.0482, reflecting respective increases of 4.04%, 5.76%, and 5.70%. Additionally, Recall@10, 20, and 50 reach 0.0566, 0.0840, and 0.1469, with improvements of 6.60%, 1.31%, and 6.29%, respectively. On the MovieLens dataset, our model achieves a Precision@20 of 0.0210, outperforming the best baseline by 12.09%. The NDCG@10, 20, and 50 scores are 0.1376, 0.1767, and 0.2165, representing gains of 10.08%, 8.54%, and 16.01%, respectively. For Recall@10, 20, and 50, our model scores 0.2645, 0.3891, and 0.6036, with corresponding improvements of 1.61%, 6.81%, and 14.30%. We have derived a more comprehensive analysis and insights from the following three perspectives.

- **Accurate Enhancement of Auxiliary Information.** Enhancing auxiliary information through RAG and LLMs enables accurate, flexible, and semantically rich information integration. Unlike VBPR (He & McAuley, 2016), which solely incorporates image data, and NGCF (Wang et al., 2019), which does not leverage auxiliary information, models such as MMSSL (Wei et al., 2023) and MICRO (Zhang et al., 2022) employ multiple types of auxiliary information. Our method utilizes a search-based approach to obtain additional information, extracting and refining rich textual data to derive accurate auxiliary information.

- **Associate Auxiliary Information with Users.** Our approach improves the Recall@10 metric on the Netflix and MovieLens datasets by 6.60% and 1.61%, respectively. Although methods like LATTICE (Zhang et al., 2021) and LLMRec (Wei et al., 2024) enhance recommendation systems by incorporating side information, they often overlook the connection between augmented information and the user. Integrating product and user information across multiple dimensions more fully captures user intent. We integrate multidimensional movie information to more accurately represent user preferences, thereby maximizing the impact of supplementary information. Remarkably, our model effectively extracts user preferences even in the absence of basic user information.

- **User Preference Mining.** The effective mining of user preferences is crucial for improving recommendation performance. Our approach demonstrates significant improvements in accuracy and NDCG metrics. For instance, compared to methods such as MMSSL (Wei

et al., 2023), which address data sparsity through self-supervised signals, our method improves NDCG@10 by 0.0059 on Netflix dataset and accuracy@20 by 0.0040 on the MovieLens dataset. This improvement is attributed to our approach, which, while leveraging auxiliary information, does not rely on basic user information. Instead, it generates user preference features directly from extended information, avoiding the noise introduced by secondary usage of large models, as seen in LLMRec (Wei et al., 2024).

In summary, our ER^{2}ALM model achieves significant improvements by accurately integrating rich auxiliary information, effectively associating it with user profiles, and directly mining user preferences from extended data. These strategies collectively enhance recommendation accuracy, outperforming baselines on both Netflix and MovieLens datasets.

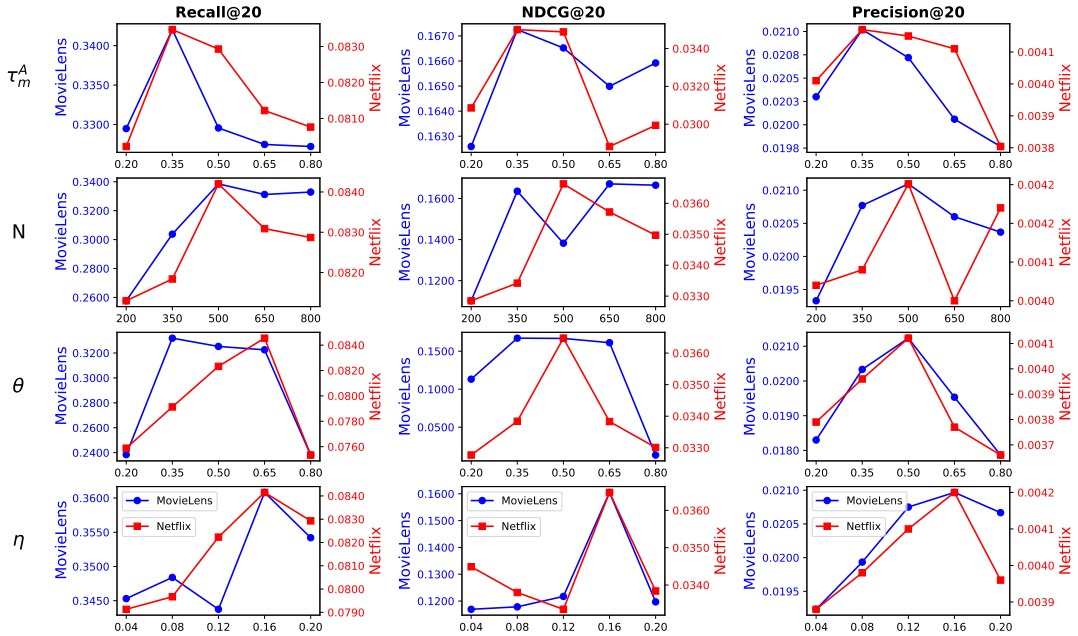

Figure 4: Performance results for the four parameters: Temperature Hyperparameter τ_{m^A}, Number of Candidate Judgments N, Confidence Threshold θ, Enhancement Factor η. The three columns of images correspond to the evaluation metrics: Recall@20, NDCG@20, and Precision@20, respectively. The vertical axes on the left and right sides represent the results for the MovieLens and Netflix datasets, respectively.

Table 4: Ablation study on key components (*i.e.*, data augmentation strategies, denoised data robustification mechanisms)

	Metrics	R@10	N@10	R@20	N@20	R@50	N@50	P@20
Aug	w/o-i	0.0396	0.0185	0.0694	0.0261	0.1306	0.0381	0.0034
	w/o-u	0.0390	0.0180	0.0683	0.0253	0.1290	0.0371	0.0033
	w/o-w	0.0482	0.0241	0.0771	0.0310	0.1388	0.0430	0.0039
Denoise	w/o noise	0.0520	0.0251	0.0786	0.0314	0.1447	0.0440	0.0039
	w/o cos	0.0543	0.0274	0.0822	0.0338	0.1387	0.0456	0.0041
	ER^{2}ALM	**0.0566**	**0.0283**	**0.0840**	**0.0367**	**0.1469**	**0.0482**	**0.0042**

Aug refers to data augmentation operations, and **Denoise** refers to the denoised data robustification mechanism.

Table 5: Analysis of temperature τ and *top-p* ρ.

Para.	Temperature τ			Top-p ρ		
Metrics	$\tau = 0.4$	$\tau = 0.8$	$\tau = 1$	$\rho = 0.6$	$\rho = 0.8$	$\rho = 1$
Netflix						
R@10	0.0531 ↓	**0.0566**	0.0537 ↓	0.0526 ↓	**0.0566**	0.0569 ↑
R@20	0.0824 ↓	**0.0840**	0.0841 ↑	0.0843 ↑	**0.0840**	0.0862 ↑
R@50	0.1312 ↓	**0.1469**	0.1436 ↓	0.1323 ↓	**0.1469**	0.1350 ↓
MovieLens						
N@10	0.1331 ↓	**0.1376**	0.1337 ↓	0.1345 ↓	**0.1376**	0.1341 ↓
N@20	0.1671 ↓	**0.1767**	0.1647 ↓	0.1659 ↓	**0.1767**	0.1686 ↓
N@50	0.2301 ↑	**0.2165**	0.2312 ↑	0.2310 ↑	**0.2165**	0.2253 ↑

4.3 Ablation and Validity Analysis (RQ2)

To evaluate the functionality and impact of each component in ER^{2}ALM, we conducted a series of ablation experiments, as shown in Table 4. We designed two main categories of experiments focusing on the augmented information and noise reduction modules. For the augmented information module, we examined the effects of (1) removing movie information, (2) removing user profile data, and (3) removing personalized user weights. For the noise reduction module, we evaluated (1) the overall noise reduction module and (2) the cosine similarity noise filtering component.

On the Netflix dataset, we observe that the absence of movie information leads to substantial performance declines. For instance, Recall@10 decreases to 0.0396, and Precision@20 reduces to 0.0034. The absence of user information leads to even worse outcomes, with NDCG@20 dropping to 0.0253 and Precision@20 reaching a low of 0.0033. Additionally, the absence of personalized user weights also yields suboptimal results, as shown by Recall@50 decreasing to 0.1388 and Precision@20 to 0.0039. These findings indicate that, due to the lack of auxiliary information, relying solely on basic movie information is insufficient for accurately analyzing user preferences, which degrades the effectiveness of user configuration information and adversely affects overall model performance. Conversely, removing sufficiently analyzed and comprehensive user information severely impacts the model's capacity to perceive and interpret user preferences, leading to further performance detriments. Lastly, omitting user-specific weight distinctions weakens the model's ability to capture user-specific details, causing suboptimal results, although retaining basic user information still ensures the model's fundamental performance.

The absence of a denoising module results in a decline in model performance, with NDCG@10 decreasing to 0.0251 and NDCG@50 to 0.0440. When only the secondary discrimination step—designed to re-evaluate noise accumulation—is removed from the denoising module, performance sees a slight improvement, with NDCG@10 reaching 0.0274 and NDCG@50 reaching 0.0456, nearly aligning with the model's optimal results. This analysis suggests that, without a filtering module for generated information, low-quality auxiliary data introduces noise, ultimately degrading the model's post-training performance. Furthermore, if the accumulated noise similarity assessment module does not perform secondary evaluation on ambiguous embeddings, residual noise may impact the final result. These findings validate the effectiveness of our denoising module.

4.4 Hyperparameter Analysis (RQ3)

In this part, we analyze several parameters within the model, examining and evaluating the impact of different parameter values through experimental observation. The parameters analyzed include: LLMs temperature τ and top-p ρ, as shown in Table 5; temperature hyperparameter τ_{m^A}, number of candidate judgments N, confidence threshold θ and enhancement factor η, as illustrated in Fig.4.

- **LLMs Temperature τ:** We observe that as the temperature parameter τ increases from 0.4, the results on both datasets initially improve, reaching a peak before slightly declining at $\tau = 1.0$. This trend can be attributed to the role of the temperature τ in controlling the randomness of generated text. Higher values of τ (e.g., $\tau = 1.0$) encourage diversity and creativity in outputs, whereas lower values ($0 < \tau < 0.1$) yield more deterministic and focused results.

- **LLMs Top-p ρ:** Lower ρ values prioritize selecting the most likely tokens, while higher values promote greater diversity by sampling from a wider range of options. We experimented with ρ values from $\{0.6, 0.8, 1\}$, and results demonstrate that higher values of top-p tend to yield better performance, likely due to RAG's ability to generate diverse and accurate auxiliary information. This diversity enables a more comprehensive, multidimensional analysis of user preferences, resulting in a more robust user preference model.

- **Temperature Hyperparameter τ_{m^A}:** The confidence threshold helps minimize the impact of low-quality generated information on model performance. We tested values within the range of {0.2, 0.35, 0.5, 0.65, 0.8}, and found that 0.4 yielded the best results. Both excessively high and low thresholds lead to suboptimal outcomes. A low threshold fails to filter out low-quality information, resulting in poor performance, whereas a high threshold discards too much information, reducing the effectiveness of subsequent processes.

- **Number of Candidate Judgments N:** The value of N influences the timing of the cosine similarity-based intervention in the reliability judgment process during noise reduction. We tested {200, 350, 500, 650, 800} and found that performing secondary cosine similarity judgments when both sets reach 500 provides the best balance between intervention timing and judgment accumulation. Higher N values require waiting for more judgment results to accumulate, which can delay auxiliary decisions based on the fuzzy threshold J, but at the same time, provide a stronger judgment basis.

- **Confidence Threshold θ:** The confidence threshold θ determines the filtering quality of the generated embeddings. A higher θ retains high-quality generated information but reduces the size of the auxiliary data. We tested values of {0.2, 0.35, 0.5, 0.65, 0.8}, and the results show that both too low and too high values are not conducive to achieving optimal results. This may be because lower thresholds include more noisy information, leading to inadequate filtering, while higher thresholds leave insufficient information, resulting in incomplete logical associations.

- **Enhancement Factor η:** The enhancement factor η controls the number of additional positive and negative samples in contrastive learning. A suitable expansion can alleviate the issue of sparse interaction data. We experimented with η values of {0.04, 0.08, 0.12, 0.16, 0.2}, and the results indicate that increasing interactions improves performance, but excessive additions lead to diminishing returns. This may be due to the introduction of new interactions driven by different user motivations, which mislead the accuracy of user analysis.

Our model contains multiple parameters, each of which has a relatively small impact on model performance across its range of variation. Collectively, these parameters have limited influence on the overall effectiveness of the model.

4.5 Case Study (RQ4)

Fig.5 illustrates various prediction outcomes for users and movies. Based on the user profiles derived from LLMs analysis, we can accurately capture user preferences and their varying priorities. Leveraging these preferences and priorities allows us to effectively align product attributes for recommendations.

On the left side is the user's profile information and preference weights, while the right side displays the movie attribute information. The connecting lines indicate the degree of alignment between the user's preferences and the movie attributes. Higher values signify a greater level of satisfaction with the user's preferences provided by the movie. In Fig.5(a), the attributes of the movie align well with the user's diverse preferences, resulting in a

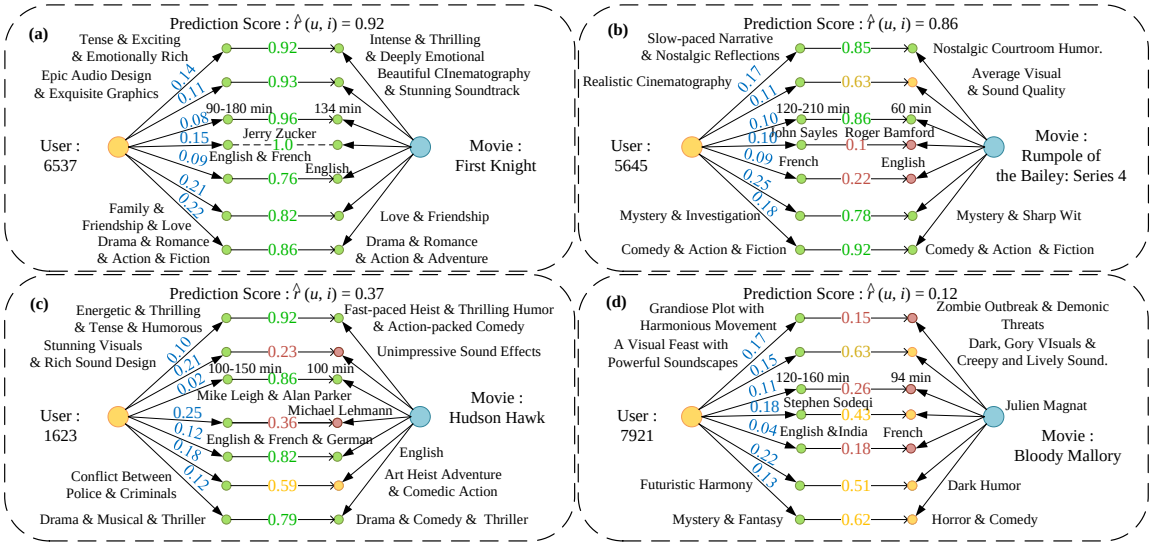

Figure 5: Examples of successful and unsuccessful user-movie recommendations: (a) and (b) illustrate successful cases, while (c) and (d) show unsuccessful ones.

high recommendation score. In Fig.5(b), although several movie attributes (e.g., director and cultural and language) do not fully match the user's preferences, the movie satisfies the user's needs in other areas (e.g., plot and theme) of higher importance, leading to a relatively high prediction score. Conversely, Fig.5(c) illustrates a scenario where the movie attributes fail to consistently address the user's key concerns. While some attributes, such as plot, cultural and language, and genre, align with the user's preferences, the preference weighting indicates that the user places greater emphasis on attributes like visual sound characteristics, the director, and the movie's theme. As these more critical preferences are not met, the movie cannot be considered a successful recommendation. Lastly, Fig.5(d) depicts a case where none of the movie attributes adequately match the user's preferences, resulting in the lowest recommendation score. This movie is clearly not aligned with the user's interests.

5. Conclusion

In this paper, we propose an Enhanced Recommendation System with Retrieval-Augmented Large Language Model (ER^{2}ALM) approach, which leverages the Retrieval-Augmented Generation (RAG) strategy to enhance the expressive power of LLMs for better capturing user preferences and interests. Specifically, we adopt a denoising strategy tailored for RAG to strengthen the retrieval and generation capabilities of LLMs, thereby effectively capturing implicit user intentions. Extensive experiments conducted on two real-world datasets demonstrate the effectiveness, accuracy, and robustness of our proposed method.

In future work, we will explore integrating domain knowledge and context-awareness to improve the framework's adaptability in diverse scenarios, further advancing its recommendation performance under cold-start and data sparsity conditions. LLMs are limited in

processing user interaction records, and there is potential to further optimize the extraction of user preferences as the quantity of satisfactory interactions increases. Additionally, the challenge of extracting user preferences becomes more pronounced as the volume of product attribute information grows, leading to a decrease in the accuracy of the derived user preferences. This issue, stemming from the increased complexity of preference extraction with an expanded information base, warrants further investigation.

Acknowledgments

This work was supported by the national key research and development program "Industrial Software" key special project "Collaborative Optimization and Dynamic Game Decision Making of Parts Supply Chain Product Service Life Cycle Process" under Grant 2022YFB3305602, National Natural Science Foundation of China under Grant 62272198 and the Humanities and Social Sciences Planning Fund of the Ministry of Education under Grant 22YJAZH110, National Natural Science Foundation of China (No.62272198, 62276277), Guangdong Key Laboratory of Data Security and Privacy Preserving (No. 2023B1212060036), Guangdong-Hong Kong Joint Laboratory for Data Security and Privacy Preserving (No. 2023B1212120007), Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515010121), Outstanding Innovative Talents Cultivation Funded Programs for Doctoral Students of Jinan University (No.2023CXB022), and this work was supported by the Jinan University.

References

Brinkmann, A., Shraga, R., Der, R. C., & Bizer, C. (2023). Product information extraction using chatgpt..

Burashnikova, A., Maximov, Y., Clausel, M., Laclau, C., Iutzeler, F., & Amini, M.-R. (2021). Learning over no-preferred and preferred sequence of items for robust recommendation. *Journal of Artificial Intelligence Research*, *71*, 121–142.

Caro-Martínez, M., Jiménez-Díaz, G., & Recio-García, J. A. (2021). Conceptual modeling of explainable recommender systems: an ontological formalization to guide their design and development. *Journal of Artificial Intelligence Research*, *71*, 557–589.

Chen, H., Li, Y., Shi, S., Liu, S., Zhu, H., & Zhang, Y. (2022a). Graph collaborative reasoning. In *Proceedings of WSDM*, pp. 75–84.

Chen, Y., Liu, Z., Li, J., McAuley, J., & Xiong, C. (2022b). Intent contrastive learning for sequential recommendation. In *Proceedings of Web Conference*, pp. 2172–2182.

Christakopoulou, K., Lalama, A., Adams, C., Qu, I., Amir, Y., Chucri, S., Vollucci, P., Soldo, F., Bseiso, D., Scodel, S., Dixon, L., Chi, E. H., & Chen, M. (2023). Large language models for user interest journeys..

Di Palma, D. (2023). Retrieval-augmented recommender system: Enhancing recommender systems with large language models. In *Proceedings of RecSys*, pp. 1369–1373.

Doddapaneni, S., Sayana, K., Jash, A., Sodhi, S., & Kuzmin, D. (2024). User embedding model for personalized language prompting..

Fatemi, B., Halcrow, J., & Perozzi, B. (2024). Talk like a graph: Encoding graphs for large language models. In *Proceedings of ICLR*.

Fu, Z., Li, X., Wu, C., Wang, Y., Dong, K., Zhao, X., Zhao, M., Guo, H., & Tang, R. (2023). A unified framework for multi-domain ctr prediction via large language models. In *Proceedings of Web Conference*, p. TBD.

Gao, Y., Sheng, T., Xiang, Y., Xiong, Y., Wang, H., & Zhang, J. (2023). Chat-rec: Towards interactive and explainable llms-augmented recommender system. In *Proceedings of Web Conference*, p. TBD.

He, J., Qiu, W., Zhuo, S., Xu, M., Zhang, Q., Xiong, Z., & Zheng, Z. (2024). Example paper title. *Journal of AI Research*, *58*, 112–134.

He, R., & McAuley, J. (2016). Vbpr: visual bayesian personalized ranking from implicit feedback. In *Proceedings of AAAI*, Vol. 30.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the SIGIR*, pp. 639–648.

Hou, Y., He, Z., McAuley, J., & Zhao, W. X. (2023). Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of ACM Web*, pp. 1162–1171.

Hou, Y., Mu, S., Zhao, W. X., Li, Y., Ding, B., & Wen, J.-R. (2022). Towards universal sequence representation learning for recommender systems. In *Proceedings of SIGKDD*, pp. 585–593.

Koto, F., Baldwin, T., & Lau, J. H. (2022). Ffci: A framework for interpretable automatic evaluation of summarization. *Journal of Artificial Intelligence Research*, *73*, 1553–1607.

Le, D. D., & Lauw, H. (2021). Efficient retrieval of matrix factorization-based top-k recommendations: A survey of recent approaches. *Journal of Artificial Intelligence Research*, *70*, 1441–1479.

Li, C., Ge, Y., Mao, J., Li, D., & Shan, Y. (2023). Taggpt: Large language models are zero-shot multimodal taggers..

Li, Y., Ma, S., Wang, X., Huang, S., Jiang, C., Zheng, H.-T., Xie, P., Huang, F., & Jiang, Y. (2024). Ecomgpt: Instruction-tuning large language models with chain-of-task tasks for e-commerce. In *Proceedings of AAAI*, Vol. 38, pp. 18582–18590.

Liao, J., Li, S., Yang, Z., Wu, J., Yuan, Y., Wang, X., & He, X. (2024). Llara: Large language-recommendation assistant..

Liu, F., Liu, Y., Chen, H., Cheng, Z., Nie, L., & Kankanhalli, M. (2024a). Understanding before recommendation: Semantic aspect-aware review exploitation via large language models..

Liu, Z., Chen, Z., Zhang, M., Duan, S., Wen, H., Li, L., Li, N., Gu, Y., & Yu, G. (2024b). Modeling user viewing flow using large language models for article recommendation. In *Proceedings of Web Conference*, pp. 83–92.

Lu, Y., Bao, J., Song, Y., Ma, Z., Cui, S., Wu, Y., & He, X. (2021). Revcore: Review-augmented conversational recommendation..

Lyu, H., Jiang, S., Zeng, H., Xia, Y., Wang, Q., Zhang, S., Chen, R., Leung, C., Tang, J., & Luo, J. (2024). Llm-rec: Personalized recommendation via prompting large language models..

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Proceedings of NeurIPS*, *32*.

Ren, X., Wei, W., Xia, L., Su, L., Cheng, S., Wang, J., Yin, D., & Huang, C. (2024a). Representation learning with large language models for recommendation. In *Proceedings of Web Conference*, pp. 3464–3475.

Ren, X., Xia, L., Yang, Y., Wei, W., Wang, T., Cai, X., & Huang, C. (2024b). Sslrec: A self-supervised learning framework for recommendation. In *Proceedings of the WSDM*, WSDM '24, p. 567–575. ACM.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). Bpr: Bayesian personalized ranking from implicit feedback..

Shi, K., Sun, X., Wang, D., Fu, Y., Xu, G., & Li, Q. (2024). Llama-e: Empowering e-commerce authoring with object-interleaved instruction following..

Tanjim, M. M., Su, C., Benjamin, E., Hu, D., Hong, L., & McAuley, J. (2020). Attentive sequential models of latent intent for next item recommendation. In *Proceedings of Web Conference*, pp. 2528–2534.

Tian, C., Xie, Y., Li, Y., Yang, N., & Zhao, W. X. (2022). Learning to denoise unreliable interactions for graph collaborative filtering. In *Proceedings of the SIGIR*, pp. 122–132.

Torbati, G. H., Tigunova, A., Yates, A., & Weikum, G. (2023). Recommendations by concise user profiles from review text..

Wang, H., Xu, Y., Yang, C., Shi, C., Li, X., Guo, N., & Liu, Z. (2023). Knowledge-adaptive contrastive learning for recommendation. In *Proceedings of WSDM*, pp. 535–543.

Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. In *Proceedings of the SIGIR*, pp. 165–174.

Wei, W., Huang, C., Xia, L., Xu, Y., Zhao, J., & Yin, D. (2022). Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of WSDM*, pp. 1120–1128.

Wei, W., Huang, C., Xia, L., & Zhang, C. (2023). Multi-modal self-supervised learning for recommendation. In *Proceedings of Web Conference*, pp. 790–800.

Wei, W., Ren, X., Tang, J., Wang, Q., Su, L., Cheng, S., Wang, J., Yin, D., & Huang, C. (2024). Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of WSDM*, pp. 806–815.

Wei, Y., Wang, X., He, X., Nie, L., Rui, Y., & Chua, T.-S. (2021a). Hierarchical user intent graph network for multimedia recommendation. *IEEE Transactions on Multimedia*, *24*, 2701–2712.

Wei, Y., Wang, X., Li, Q., Nie, L., Li, Y., Li, X., & Chua, T.-S. (2021b). Contrastive learning for cold-start recommendation. In *Proceedings of ACM MM*, pp. 5382–5390.

Wei, Y., Wang, X., Nie, L., He, X., & Chua, T.-S. (2020). Graph-refined convolutional network for multimedia recommendation with implicit feedback. In *Proceedings of ACM MM*, pp. 3541–3549.

Wei, Y., Wang, X., Nie, L., He, X., Hong, R., & Chua, T.-S. (2019). Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of ACM MM*, pp. 1437–1445.

Wu, D., Zhuo, S., Wang, Y., Chen, Z., & He, Y. (2023). Online semi-supervised learning with mix-typed streaming features. In *Proceedings of AAAI*, pp. 4720–4728. AAAI Press.

Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., & Xie, X. (2021). Self-supervised graph learning for recommendation. In *Proceedings of SIGIR*, pp. 726–735.

Wu, J., Chang, C.-C., Yu, T., He, Z., Wang, J., Hou, Y., & McAuley, J. (2024). Coral: Collaborative retrieval-augmented large language models improve long-tail recommendation. In *Proceedings of SIGKDD*, pp. 3391–3401.

Xi, Y., Liu, W., Lin, J., Cai, X., Zhu, H., Zhu, J., Chen, B., Tang, R., Zhang, W., & Yu, Y. (2024). Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of RecSys*, pp. 12–22.

Yin, B., Xie, J., Qin, Y., Ding, Z., Feng, Z., Li, X., & Lin, W. (2023). Heterogeneous knowledge fusion: A novel approach for personalized recommendation via llm. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pp. 599–601.

Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., & Nguyen, Q. V. H. (2022). Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of SIGIR*, pp. 1294–1303.

Zhang, J., Zhu, Y., Liu, Q., Wu, S., Wang, S., & Wang, L. (2021). Mining latent structures for multimedia recommendation. In *Proceedings of ACM MM*, pp. 3872–3880.

Zhang, J., Zhu, Y., Liu, Q., Zhang, M., Wu, S., & Wang, L. (2022). Latent structure mining with contrastive modality fusion for multimedia recommendation. *IEEE Transactions on Knowledge and Data Engineering*, *35*(9), 9154–9167.

Zhao, X., Hu, B., Zhong, Y., Huang, S., Zheng, Z., Wang, M., & Wang, H. (2025). Raserec: Retrieval-augmented sequential recommendation. In *Proceedings of Web Conference*, p. TBD.

Zheng, Z., Chao, W., Qiu, Z., Zhu, H., & Xiong, H. (2024). Harnessing large language models for text-rich sequential recommendation. In *Proceedings of Web Conference*, pp. 3207–3216.

Zhou, X., Hu, Z., Huang, J., & Chen, J. (2023). Decentralized gradient-quantization based matrix factorization for fast privacy-preserving point-of-interest recommendation. *Journal of Artificial Intelligence Research*, *76*, 1019–1041.

Zhu, Y., Wu, L., Guo, Q., Hong, L., & Li, J. (2024). Collaborative large language model for recommender systems. In *Proceedings of Web Conference*, pp. 3162–3172.

Zhuo, S., Qiu, J.-J., Wang, C.-D., & Huang, S.-Q. (2024a). Online feature selection with varying feature spaces. In *Proceedings of IEEE International Conference on Knowledge and Data Engineering*, pp. 4806–4819.

Zhuo, S., Wu, D., Hu, X., & Wang, Y. (2024b). Ardst: An adversarial-resilient deep symbolic tree for adversarial learning. *International Journal of Intelligent Systems*, *2024*(1), 2767008.